Amendments to the Specification:

Please replace the paragraph beginning on page 2, line 25 with the following amended paragraph:

For example, different suppliers of web browsers may incorporate root CA certificates issued by many different sources. Each of these sources may issue a certificate with differing expiration dates. Management of the root CA certificates by a trusted authority is typically not used. Accordingly, a problem arises when different versions of web browsers are used by different users For example, an older version of a web browser may have root CA certificates that expire sooner than root CA certificates that may be embedded in newer versions of web browsers. Accordingly, various certificate issuing entities may serve as different root CA's and issue certificates having differing expiry periods. When a root CA certificate expires, all servers which have web certificates that were issued by that CA will no longer ~~be trusted by~~ trust any browser which contains only the expired certificate for that CA.

Please replace the paragraph beginning on page 3, line 6 with the following amended paragraph:

For a conventional web model, there is typically no way to detect the expiration of a web certificate prior to a request for a session with a web server. For example, web certificates that are preinstalled with web browsers from different issuers are typically not continually checked by the web browser to insure that they have not expired. Typically, a user will only be informed of a problem when the web browser attempts to set up a secure session with a web server. If the web certificate has expired, the session is not granted. One proposed solution has been to require a user to manually update a web browser that has prestored web certificates that expire at later dates. Typically, web servers will detect old web browser versions through, for example, web identification tags embedded in headers and identify a link (e.g., URL) to the site that may contain a new version of a web [[server]]browser. The user then typically clicks on a URL to connect to the site containing the new software version and downloads the new web browser containing web certificates with expiry periods later than those on previous web browser versions.

Please replace the paragraph beginning on page 3, line 27 with the following amended paragraph:

However, a problem arises with such techniques since, inter alia, a user typically is denied a secure session and is additionally required to manually obtain an ~~upgrade~~ updated version of a web [[server]]browser. Accordingly, when a user installs a new version of a web browser it is typically not possible for a web site to know that the web browser has the new root CA certificate without establishing an SSL connection or other suitable secure session that requires the use of a new root CA certificate. This problem can be overcome by issuing a cookie to the user's browser. The next time the user visits the site, the server can check for the cookie. If the cookie exists, the server knows that the user has installed the new root CA certificate. However, other sites that also require the new root CA certificate cannot read that cookie. As such, each different server in a different domain may not be able to identify that the user has already installed the new root CA certificate.

Please replace the paragraph beginning on page 5, line 10 with the following amended paragraph:

For example in an embodiment applied to a system employing web certificates, the web browser contacts the web server, the web server, upon detecting an unsuitable version of the web browser, notifies the web browser to go obtain new embedded web certificates. Accordingly, the web server automatically redirects the web browser to a third server such as a software update control server. The software update controller contains the latest version of the software, root CA certificate, or other data required by the web server. The web browser obtains a cookie from the software update controller, as well as a message for the web server embedded in an URL. The message in the URL from the software update controller is detected by the web server so that the web server 1) can issue [[it's]]its own cookie to the browser to indicate that the ~~upgrade~~ update has been complete and 2) trusts that the web browser has the unexpired web certificate or other updated data.

Please replace the paragraph beginning on page 5, line 22 with the following amended paragraph:

The systems and methods may be employed to update the software in different versions, provide unexpired root CA certificates, or provide any other suitable data. The system allows a user that has updated the root CA certificates to connect to a different site after ~~upgrading~~ updating wherein the different site detects if the data has already been ~~upgraded~~ updated or a new CA certificate downloaded to a web browser by detecting the ~~universal~~ cookie from the software update controller. For example, a first time through, a user manually inserts the new root CA certificate in the web browser. The next time the user accesses a site that is in the program, it will be automatic. ~~The different web servers cannot typically detect a 'universal cookie' of any sort.~~ The browser gets a cookie and a special message encoded in the URL, or inserted into the HTTP headers, from the software update controller. The web server detects the special message in the URL or the HTTP headers, not in the cookie. The webserver then sets its own cookie for identification at a later date.

Please replace the paragraph beginning on page 6, line 5 with the following amended paragraph:

FIG. 1 illustrates a system 100 for updating data that includes first processing entities ~~102a-1-2n~~102a-102n, such as devices containing a web browser, second processing entities 104a-104n, such as web servers, and a third processing entity 106, such as a software update controller (e.g., another server). The third processing entity 106 is preferably in operative communication with only the first processing entities 102a-102n. Also in a preferred embodiment, the plurality of second processing entities 104a-104n are in operative communication with the first processing entities 102a-102n but are not in communication with the third processing entity 106. For purposes of illustration and not limitation, the disclosed invention will be described with reference to an Internet-based system that employs web certificates as the data to be updated. However, it will be recognized that the invention may be applicable to any suitable information security system such as wireless communication systems, intranet based systems, any systems requiring updating of versions of software, or any other suitable system.

Please replace the paragraph beginning on page 8, line 1 with the following amended paragraph:

If the connection request does not include the appropriate cookie (cookieswuc) for the software update controller 106, the software update controller 106 recognizes that this may be the first update request required by the first processing entity. Where the redirected connection request 116 does not include the cookie of the destination processing entity, the third processing entity sends update instructions 118 to the first processing entity along with a request for a confirmation of completion of an update. This may be done, for example, by requesting the user to activate a GUI interface confirmation of update button. The update instructions, as shown in block 216, may include, for example, instructions to be displayed for the user to select which version of the software to update to or which web certificates should be embedded in the web browser and whether the new version of the web browser, or other data from the third processing entity was received by the first processing entity. Accordingly, the third processing entity causes the first processing entity to display instructions for the user to follow so that the appropriate version of the software is updated or provided to the first processing entity. The user then selects the confirmation button to indicate that the version has been selected and an update has been completed. This update confirmation data 120 is then sent from the first processing entity to the third processing entity in response to receiving the request for confirmation of the completion of an update. The update confirmation data 120 may include, for example, the URL of the third processing entity (URLswuc), a header with the return address of the second processing entity, and ~~upgrade~~ update complete data indicating that the ~~upgrade~~ update has been completed. The update instruction includes the new version of the software which may be communicated in any suitable form, such as encrypted using a public key encryption engine, symmetric key encryption engine or any other suitable encryption technique.

Please replace the paragraph beginning on page 8, line 27 with the following amended paragraph:

As shown in block 218, the third processing entity receives the update confirmation data 120 and parses the header to verify that the ~~upgrade~~ update is complete. More particularly, the third processing entity checks to detect that the update complete data is included in the update confirmation data indicating that the first processing entity has properly received and suitably ~~upgraded~~ updated its web certificates, software, or other data in accordance with the update instruction 118. The third processing entity parses the header, for example, to see that the

~~upgrade~~ update complete data and that the cookie associated with the software update controller for that particular update has been set in the first processing entity. The third processing entity therefore sets the cookie in the first processing entity. The third processing entity then sends an update complete and redirect command 122 back to the first processing entity, for detection by the second processing entity. This update complete data and redirect command 122 contains, for example, a redirect command back to the second processing entity which may include, for example, the URL of the second processing entity along with data representing that the third processing entity cookie has been set in the first processor. As shown in block 220, the first processor generates another connection request 124 to the second processor indicating that the software update is complete. For example, this includes the URL of the second processing entity, and a header with the data software cookie set equal "yes" as provided by the third processing entity. As shown in block 222, the second processing entity receives the connection request 124, parses the (as noted above, this information may be in the URL, or in the headers, depending on the type of CGI request - POST or GET) header to detect the cookieswuc set equal yes, and then sets the cookie of the second processing entity in the first processing entity through communication 126. The process continues as needed for other processing entities and other second processing entities, as desired.

Please replace the paragraph beginning on page 9, line 22 with the following amended paragraph:

As applied to a system requiring web certificates, the first processing entity is a web browser that is operative to request a connection with the web server 104a. The web server 104a detects a need to update web certificate data based on the request for a connection from the web browser by determining, for example, that no cookie associated with the software update controller 106 has been provided to the web browser [[104a]]102a. The web browser 104a automatically redirects communication from the web browser 104a and the web server, to the web browser and the web certificate update controller in response to detecting the need to update the web certificate. The web server, for example, sends the universal resource locator associated with the web certificate update controller, and other information, as desired, to automatically force the first processing entity to communicate with the software update controller. The software update controller 106, may be a web certificate update controller that contains new versions of web browsers that contain web certificates having later expiry periods, for example. The web certificate update controller provides web certificate update complete data 122 for the web server through the web browser.